

## ASSOCIATION OF DATA WITH A PRODUCT CLASSIFICATION SCHEMA

## TECHNICAL FIELD OF THE INVENTION

This invention relates generally to electronic commerce and more particularly to association of data with a product classification schema.

## BACKGROUND OF THE INVENTION

Due to the ever-increasing popularity and accessibility of the Internet as a medium of communication, the number of business transactions conducted using the Internet is also increasing, as are the numbers of buyers and sellers participating in electronic marketplaces providing a forum for these transactions. The majority of electronic commerce ("e-commerce") transactions occur when a buyer determines a need for a product, identifies a seller that provides that product, and accesses the seller's web site to arrange a purchase of the product. If the buyer does not have a preferred seller or if the buyer is purchasing the product for the first time, the buyer will often perform a search for a number of sellers that offer the product and then access numerous seller web sites to determine which seller offers certain desired product features at the best price and under the best terms for the buyer. The matching phase of e-commerce transactions (matching the buyer with a particular seller) is often inefficient because of the large amount of searching involved in finding a product and because once a particular product is found, the various offerings of that product by different sellers may not be easily compared.

SUMMARY OF THE INVENTION

According to the present invention, disadvantages and problems associated with previous data identification and association techniques have been substantially reduced or eliminated.

In one embodiment of the present invention, a computer-implemented system for associating target data with a product classification schema includes a data association module that accesses the product classification schema. The schema includes a taxonomy that includes a hierarchy of classes into which products may be categorized. The schema further includes ontologies that are associated with one or more of the classes. Each ontology includes one or more product attributes. The data association module accesses the target data to be associated with the schema and determines one or more classes with which at least a portion of the target data should be associated. This determination is based on a comparison between the target data and the product attributes of the ontologies or between the target data and values for one or more of the product attributes. Furthermore, the data association module associates at least a portion of the target data with one or more classes in response to determining one or more classes with which at least a portion of the target data should be associated.

Particular embodiments of the present invention may provide one or more technical advantages. For example, certain embodiments of the present invention may be used in association with a global content directory that categorizes a number of different products and provides a portal through which a buyer may search for particular products and establish communications with an appropriate seller of a desired product. The global content directory may use one or more schema to categorize the various products. Each schema includes a taxonomy, which is a hierarchy of classes into which the products may be categorized. Furthermore, one or more of the classes included in the taxonomy may have an associated ontology, which includes one or more attributes associated with a product or a seller of a product.

Product data may be generated for use with the global content directory. Such data may be in a format appropriate for the global content directory and identified for use with the global content directory. For example, the data may be organized according to the ontologies of the various classes of the global content directory.

However, it may be desirable to associate product data with the global content directory even though that data is not clearly associated with such ontologies. Therefore, certain embodiments of the present invention provide a data association module that identifies product data to be associated with the global content directory and properly associates the data with classes of the global content directory based on the content of the data. Thus, although particular data may not be optimally created and organized for the global content directory, embodiments of the present invention may be used to identify and associate the data with appropriate classes of the global content directory. This data association allows existing product data to be associated with the global content directory without the expense of generating new data or modifying existing data. Furthermore, this existing data, at least in part, is properly associated with the global content directory so that a buyer searching for products using the global content directory can effectively access the product data through the classes of the global content directory.

Other technical advantages may be readily apparent to those skilled in the art from the figures, description, and claims included herein.

020431.0843

BRIEF DESCRIPTION OF THE DRAWINGS

To provide a more complete understanding of the present invention and the features and advantages thereof, reference is made to the following description taken in conjunction with the accompanying drawings, in which:

- 5 FIGURE 1 illustrates an example electronic commerce system;
- FIGURE 2 illustrates an example directory structure of an example global content directory;
- FIGURE 3 illustrates an example table of a seller database;
- 10 FIGURE 4 illustrates an example portion of a schema including a taxonomy and product ontology and an example portion of a schema including only a taxonomy;
- FIGURE 5 illustrates an example method for translating between different schemas;
- FIGURE 6 illustrates an example method for associating product data with a schema; and
- 15 FIGURE 7 illustrates an example electronic commerce system in further detail.

P23353250

DESCRIPTION OF EXAMPLE EMBODIMENTS

FIGURE 1 illustrates an example system 10 that includes a network 12 coupling buyers 20, sellers 30, and a global content directory (GCD) server 40. System 10 enables electronic commerce (“e-commerce”) transactions between buyers 20 and sellers 30 through the use of a GCD 42 supported by GCD server 40. GCD 42 may be internal or external to GCD server 40. Network 12 may include any appropriate combination of public and/or private networks coupling buyers 20, sellers 30, and GCD server 40. In an example embodiment, network 12 includes the Internet and any appropriate local area networks (LANs), metropolitan area networks (MANs), or wide area networks (WANs) coupling buyers 20, sellers 30, and GCD server 40 to the Internet. Since the Internet is accessible to the vast majority of buyers and sellers in the world, the present invention potentially includes all of these buyers and sellers as buyers 20 and sellers 30 associated with system 10. However, the use of the term “global” should not be interpreted as a geographic limitation necessarily requiring that GCD 42 provide directory services to buyers 20 and sellers 30 around the world (or in any other particular region) or that the content of GCD 42 be from all over the world (or from any other particular region).

Although buyers 20 and sellers 30 are described as separate entities, a buyer 20 in one transaction may be a seller 30 in another transaction, and vice versa. Moreover, reference to “buyer” or “seller” is meant to include a person, a computer system, an organization, or another entity where appropriate. For example, a buyer 20 may include a computer programmed to autonomously identify a need for a product, search for that product, and buy that product upon identifying a suitable seller. Although buying and selling are primarily described herein, the present invention contemplates any appropriate e-commerce transaction. Moreover, reference to “products” is meant to include goods, real property, services, information, or any other suitable tangible or intangible things.

A typical e-commerce transaction may involve a “matching” phase and a “transactional” phase. During the matching phase, a buyer 20 may search for a suitable product (meaning any good, real property, service, information, or other tangible or intangible thing that may be the subject of an e-commerce transaction) offered by one or more sellers 30, identify the most suitable seller 30 (which may

involve, for example, identifying the seller 30 offering the lowest price), and contact that seller 30 to enter the transactional phase. During the transactional phase, the buyer 20 and seller 30 may negotiate a contract for the sale of the product (which may involve, for example, more clearly defining the subject of the transaction, negotiating a price, and reaching an agreement on supply logistics) and generate a legal document embodying the terms of the negotiated contract. To identify the most suitable seller 30 during the matching phase without the use of GCD 42, a buyer 20 may have to access numerous seller web sites to determine which seller 30 offers certain desired features of the product at the best price. Sellers 30 may each provide one or more databases 32, such as relational databases, that include data identifying the products available from sellers 30 and their features. Each database 32 may be accessed through the associated seller's web site or in any other appropriate manner. The multiple one-to-one (one buyer 20 to one seller 30) searches that this process requires are inefficient and expensive because of the large amount of searching involved in finding a product and because the various offerings of that product by different sellers 30 may not be easily compared.

Alternatively, multiple sellers 30 may be grouped in an electronic marketplace according to the products they provide and a buyer 20 may search the offerings of the multiple sellers 30 at a single web site. However, if buyer 20 wishes to obtain several different types of products, buyer 20 may have to go to several different types of marketplaces. Furthermore, there may be numerous competing marketplaces that buyer 20 has to search to perform the matching phase of a transaction for a particular product. One potential method of addressing this problem is to create a global product database that potentially includes data identifying the features of all the products that any buyer may wish to obtain. Therefore, the global database would include the combined contents of every database 32 associated with every seller 30. However, such a global database would have many problems. For example, the sheer size of the database would make it difficult to search and thus the database would suffer from performance problems. In addition, it would be difficult to allow large numbers of buyers 20 to search the database at once. Furthermore, all sellers 30 would be required to access the global database to update their information and the entire

0633655252  
= 06320002

database would have to be updated each time a change is made. Many other problems might also exist.

A solution to the above problems, at least in part, is GCD 42. GCD 42 is a universal directory of the contents of multiple seller databases 32 (and potentially all seller databases 32). GCD 42 may be implemented using one or more servers 40 or other computers located at one or more locations. Most or all of the content in these seller databases 32 remains stored in databases 32, but this content is accessible using GCD 42. Therefore, like the global database described above, GCD 42 provides buyers 20 with access to product data relating to a multitude of products (and potentially seller data relating to one or more sellers 30 of the products), but unlike the global database, GCD 42 does not attempt to store all of this data in one enormous database. Where appropriate, reference to "data" is meant to include product data (meaning information reflecting values for certain attributes of a product), seller data (meaning information reflecting values for certain seller attributes), or both product data and seller data.

GCD 42 provides a directory of products using a directory structure in which products are organized using a hierarchical classification system. A buyer 20 may navigate or search the directory to find a particular product class into which products are categorized. The product data (and potentially seller data) associated with a product included in a product class may actually be stored in and obtained by GCD 42 from a seller database 32. However, the requested data may be transparently provided to buyer 20 such that all of the product data may appear to buyer 20 as being included in GCD 42. Although product and/or seller data has primarily been described as being stored in seller databases 32, the present invention contemplates product data being stored in any suitable manner and being retrieved from any suitable sources. For example, system 10 may include a shared data repository 34 that contains product data and/or seller data that may be combined with data from one or more seller databases 32, as described in further detail below.

Furthermore, as is described in further detail below with reference to FIGURES 4 and 5, system 10 may include a translation tool 36 including a mapping module 37 and an ontology generation module 38 that may be used to translate between different mechanisms used to organize the product data stored in seller

databases 32 and/or repository 34. Moreover, as is described in further detail below with reference to FIGURE 6, system 10 may include a data association module 39 that may be used to associate data in seller databases 32 or other data sources with GCD 42. Data association module 39 may be integral with or separate from translation tool 36. Furthermore, translation tool 36 and/or data association module 39 may be integral with or separate from GCD server 40.

FIGURE 2 illustrates an example directory structure 44 of an example GCD 42. Products categorized in GCD 42 may be organized according to schemas. A schema may include a set of product classes (which may be referred to as a "taxonomy") organized in a hierarchy, each class being associated with a set of product features, characteristics, or other product attributes (which may be referred to as a "product ontology"). For example, pens may have different kinds of tips (such as ball point or felt tip), different tip sizes (such as fine, medium, or broad), and different ink colors (such as blue, black, or red). Accordingly, a schema may include a class corresponding to pens that has a product ontology including *tip type*, *tip size*, and *color*, or other appropriate attributes. Within a class, products may be defined by product attribute values (such as, for example, ball point, medium tip, blue ink). Reference to "value" is meant to include any appropriate data reflecting an instance of a product attribute or a seller attribute. Product attribute values and seller attribute values may include numbers, letters, figures, characters, symbols, or other suitable information for describing a product or a seller, respectively. In one embodiment, a product ontology may be divided into entry-required attributes (meaning attributes for which a value has to be provided) and entry-optional attributes (meaning attributes for which a value is optional), and these categories may be further divided into commercial features and design features (or any other suitable divisions).

In addition to a taxonomy and product ontologies, a schema may include a set of attributes for each seller (which may be referred to as a "seller ontology"). Such attributes may include geographic restrictions (such as served markets), currencies accepted by each seller, collaboration tools accepted by each seller, contract terms accepted by each seller, types of contracts accepted by each seller, levels of buyer credit required by each seller, and any other suitable seller attributes. Similar to a products within a product class, sellers offering products within a product class may

020431.0843  
TOP SECRET

be defined by seller attribute values corresponding to seller attributes. Accordingly, a schema may include a set of classes, each including one or more products, and each class may be associated with a set of product attributes and a set of seller attributes.

In example directory structure 44, products may be organized and cataloged according to industry standard schemas 46 or other appropriate schemas, as described below. Within industry standard schemas 46, there are two example classes: a “direct material” class 48 and an “indirect material” class 50. Each of these classes 48 and 50 includes several sub-classes (which may themselves include sub-classes). Therefore, the numerous classes of directory structure 44 form a “tree-like” hierarchical structure into which products may be categorized. For example purposes, certain portions of directory structure 44 are “expanded” in FIGURE 2 to show various levels of classes. The “level” of a class is indicated by the number of other classes between that class and a root class. For example, “indirect material” class 50 is at the same level in directory structure as “direct material” class 48. “Indirect material” class 50 may include an “office and computer supplies” class 52, which includes a “desk supplies” class 54, which includes a “writing utensils” class 56. Furthermore, “writing utensils” class 56 includes a “pens” class 58, which includes numerous pen type classes 60a-60n (“n” indicating that any number of classes 60 may be included in “pens” class 58). Each of classes 50, 52, 54, 56, 58, and 60 is located at a different level of directory structure 44. A class at any level in directory structure 44 may include one or more sub-classes, those sub-classes may include one or more sub-classes, and so on until a desired specificity of categorization is reached. A series of classes from a highest level class (the broadest class) to a lowest level class (the most specific class) may be referred to as a “branch” of directory structure 44. For example, classes 46, 48, 50, 52, 54, 56, 58, and 60b form one branch of directory structure 44.

A buyer 20 may navigate through directory structure 44 by expanding or collapsing various classes as desired. For example, FIGURE 2 illustrates an expansion of certain classes of directory structure 44 to reach a “felt-tip pen” class 60b. Once a buyer 20 has navigated to a class that is specific enough for buyer 20 (and/or a “leaf” class that is at the end of a branch), buyer 20 may perform a search for products within that class. For example, buyer 20 can search for all products in “writing utensils” class 56 that are blue felt-tip pins having medium tips.

Alternatively, if buyer 20 navigates to the end of a branch of directory structure 44 (to a leaf class), such as “felt-tip pen” class 60b, GCD 42 may then enable buyer 20 to search for such pens that have blue ink and medium tips (which may reach the same result as the search above).

5       Buyer 20 may also search for sellers matching one or more seller attribute values within a product class. For example, in addition to searching for all products in writing utensils class 56 that are blue felt-tip pins having medium tips, buyer 20 may search for sellers 30 serving Texas that accept U.S. dollars. Buyer 20 may search for products matching certain product attribute values and sellers matching certain seller attribute values in any appropriate manner. In one embodiment, for example, buyer 20 provides search criteria including both values for product attributes and for seller attributes (search criteria may instead be generated automatically, in whole or in part, as described below), and server 40 searches for products that match the product attribute criteria and are offered by sellers matching the seller attribute criteria. In another embodiment, buyer 20 provides only product attribute values as search criteria, and server 40 limits its search for products matching the product attribute criteria to databases 32 associated with sellers 30 known to match seller attribute criteria that buyer 20 may want according to a buyer profile or otherwise.

20      As described above, in one embodiment product data (at least product data more detailed than data provided by a taxonomy) and seller data are not stored in GCD 42, but are stored in databases 32. For example, a seller 30 may maintain a relational database 32 that includes a plurality of tables containing product attribute values for a variety of products and seller attribute values for each product, a set of products, or all of the products offered by seller 30. Product data and seller data may be integrated into one or more tables or may be segregated into different tables. Moreover, product data and seller data for a seller 30 may be stored in the same or separate databases. One or more pointers may be associated with each class to identify the location of one or more databases 32 that include product data and/or seller data for products contained in that class or to identify particular data in databases 32. Therefore, GCD 42 may execute a search for products in databases 32 identified by a pointer corresponding to a user-selected (or automatically selected) class. GCD 42 may also return the network location (such as a uniform resource

020431.0843-06223974

locator (URL) or other network address) of the database 32 to buyer 20 so that buyer 20 may independently access database 32. Databases 32 may be searched using any appropriate method including, but not limited to, a structured query language (SQL) query.

GCD 42 may be implemented using the lightweight directory access protocol (LDAP), which enables directories to be provided using the tree-like structure described above. However, any other appropriate technique or protocol for creating GCD 42 may alternatively be used and GCD 42 may have any appropriate structure. Furthermore, GCD 42 may be an object-oriented directory (which is also provided by LDAP) such that each class in directory structure 44 includes the attributes of parent classes in which the class is a sub-class. In this embodiment, a product class listed at the end of a branch of the tree structure (a leaf class) includes all of the attributes of its parent classes in the branch. Furthermore, each product included in a database 32 may be an object that includes all the attributes of the classes in which the product is included. Thus, when a search is performed from a leaf class of directory structure 44, the search query may automatically include any appropriate attributes of parent classes of the leaf class.

For example, if a buyer 20 has navigated through directory structure 44 to “felt-tip pens” class 60b, a search performed by buyer 20 (or by GCD 42 on behalf of buyer 20) from felt-tip pens class 60b may automatically be limited to a search for felt-tip pens and buyer 20 may introduce additional desired search criteria (such as blue ink and medium tip). Therefore, if a database 32 searched includes product data relating to a variety of writing utensils, a search of database 32 may be automatically limited by GCD 42 to only include felt-tip pens within that database 32. Buyer 20 may also identify additional product attribute values and/or seller attribute values as additional search criteria.

FIGURE 3 illustrates an example table 150 that may be included in a seller database 32 and/or repository 34. Database 32 and repository 34 may include one or more tables 150, and each table 150 may contain data relating to one or more products. For example, example table 150 includes data relating to different types of pens. Table 150 may also include data for other types of products (for example, other types of office supplies), or such data may be contained in other tables 150 in

database 32 and/or repository 34. Table 150 includes a plurality of columns 152 that each include data relating to a particular product attribute or seller attribute. Although an example number of columns 152 including example product attribute values and seller attribute values are illustrated, it should be understood that any appropriate number and type of product attributes, seller attributes, or other categories of data may be included in table 150. Moreover, as described briefly above, seller data and product data may be segregated into different tables instead of being integrated into the same table as shown in example table 150.

Table 150 also includes a number of rows 154 that may each correspond to a particular product and that each include values for one or more of the product attributes and seller attributes. Each of the values (which may be numeric, textual, or in any other appropriate format) is located at the intersection of the row 154 associated with a particular product and the column 152 that includes a particular product attribute or seller attribute. Each of these intersections may be referred to as a field or cell 156 of table 150. Where seller data and product data are integrated, each row 154 may contain all of the product data and seller data for the product corresponding to that row 154. Alternatively, there may be a row or set of rows dedicated to seller data that may apply to all products offered by a seller 30 or a subset of all such products. Where seller data and product data are segregated, each row in the seller data table may correspond to a set of seller attribute values that may be linked to a set of one or more products in the product data table such that seller data for a product may be accessed when product data for that product is accessed, and vice versa.

The data in one or more columns 152 of table 150 may be indexed to increase the speed with which database reads may be conducted. For example, the fields 156 of ink color column 152d and tip size column 152e may be indexed so that a database query for a pen having a particular ink color and tip size may be quickly performed. Data in table 150 may be indexed using any appropriate database indexing technique. The typical result of such indexing is that when GCD 42 or a buyer 20 requests indexed data from a database 32 and/or repository 34, the associated database management system (or other appropriate interface to database 32 and/or repository 34) does not have to search through every field 156 in the tables 150 included in

database 32 and/or repository 34 to locate the requested data. Instead, the data may be indexed such that when a query is submitted for products having certain product attribute values and/or sellers 30 having certain seller attribute values that have been indexed, the database management system already knows the locations of such products in table 150 and may return data associated with these products without searching the entire table 150 or database 32 and/or repository 34 for the products. For example, if the ink color fields 156 and tip size fields 156 of columns 152d and 152e, respectively, are indexed, the index will typically identify the location of all products having black ink and a medium tip size.

If a query is submitted that also specifies a value of one or more non-indexed product attributes (for example, a query for pens manufactured by ABC Company, if the manufacturer fields 156 in column 152c are not indexed) and/or seller attributes, then the associated database management system may perform a search of database 32 and/or repository 34 for products that include the specified value of the one or more non-indexed attributes or seller attributes. However, such a search may be limited to the products already identified (using the index) as including specified values of indexed attributes (for example, pens having black ink and a medium tip) and/or seller attributes that are also included in the search. Therefore, the amount of time required to perform the search may be reduced even though one or more of the product attribute values or seller attribute values that are searched for are not indexed.

Returning to FIGURE 2, when GCD 42 has performed a search of the databases 32 and/or repository 34 (or particular tables thereof) identified by a pointer or pointers associated with a class that buyer 20 has selected or that has been automatically selected, GCD 42 may return product data and/or seller data associated with one or more products matching the search criteria. GCD 42 may integrate the product data and/or seller data resulting from the search into directory structure 44 so that the data appears to buyer 20 as being part of GCD 42. GCD 42 may alternatively present the results of the search in any other appropriate manner. Each product resulting from the search may be an object which is unique instance of the class in which buyer 20 is searching. Furthermore, each such object (and its location) may be uniquely identified using a numbering scheme corresponding to directory structure 44.

100-00000000-00000000

In summary, a buyer 20 may search for a product matching certain product attribute values available from a seller matching certain seller attribute values using GCD 42 and thus eliminate or reduce the need for buyer 20 to individually search numerous seller databases 32 to find the desired product available from a suitable seller. GCD 42 provides access to product and/or seller data relating to these numerous products using directory structure 44, which organizes products using a hierarchical, object-oriented classification system. Buyer 20 may navigate or search directory structure 44 to find a particular classification of products and various information associated with the products within this classification, initiate a search of databases 32 including product and/or seller data relating to a product, and then communicate with an appropriate database 32 through GCD server 40 or otherwise. Such access to vast numbers of products is provided without the requirement that all data about the products and/or sellers be stored in a global database. Instead, this data may be stored in seller databases 32 that can be readily accessed using GCD 42.

Although example directory structure 44 may use industry standard schemas 46 as described above with reference to FIGURE 2, any other appropriate schemas 62 may be used in addition to or instead of industry standard schemas 46. For example, while industry standard schemas 46 may be organized from a seller's viewpoint, other schemas 62 may be used that organize products from a buyer's viewpoint. For example, a buyer 20 may wish to furnish a kitchen of a new house with various products, such as appliances, window treatments, paint, cabinetry, plumbing, dishes, and cooking utensils. Using one schema 62, these products may be organized into a variety of unrelated classes based on certain features of the products (for example, certain kitchen appliances may be categorized in an electronics class 52 of directory structure 44 while paint may be categorized into an industrial class 52). However, another example schema 62 may categorize all such products into a home products class (which may include several classes further categorizing the products, such as a kitchen products class which includes a kitchen appliances class, which includes a refrigerator class, and so on). Therefore, the same product may be included in multiple schemas 62. These alternative schemas may be included in directory structure 44 and may be stored as a part of or separate from GCD 42.

020431.0843

Furthermore, although GCD 42 may not provide an alternative schema desired by a particular user, a schema 46 or 62 provided by GCD 42 may be translated to the alternative schema desired by the user. As described above, the schema 46 or 62 provided by GCD 42 include "rich" content in that these schemas 46 or 62 include both a taxonomy (hierarchy of product classes) and an ontology (product and/or seller attributes associated with each class). However, many commonly used schema, such as the United Nations Standard Products and Services Classification (UNSPSC) schema, include a taxonomy but do not include an ontology. Therefore, to translate a GCD schema 46 or 62 to such an "ontology-less" schema, the taxonomy of the GCD schema 46 or 62 is mapped to the taxonomy of the ontology-less schema and an ontology is created for each class in the ontology-less schema.

FIGURE 4 illustrates an example portion of a GCD schema 70 (including a taxonomy and product ontology) and an example portion of an ontology-less schema 80 (including only a taxonomy). Although a seller ontology is not associated with schema 70 in FIGURE 4, it should be understood that the following description applies equally to product and seller ontologies. The first step involved in translating schema 70 to schema 80 is to map the classes 72 of schema 70 to classes 82 of schema 80. For example, each leaf class 72 of schema 70 may be mapped to one or more classes 82 of schema 80 (multiple leaf classes 72 may be mapped to a single class 82). The process of mapping classes 72 to classes 82 may be performed by a user of system 10, such as a buyer 20, a seller 30, or a user associated with GCD server 40. The user may use mapping module 37 of translation tool 36 to associate a leaf class 72 and/or particular pointers associated with a leaf class 72 with one or more classes 82. For example, mapping module 37 may present a graphical representation of classes 72 and 82 to the user and allow the user to "drag and drop" (using a mouse or other input device) an icon representing a class 72 onto another icon representing a class 82. Multiple leaf classes 72 included in the same parent class may be mapped to a class 82 by mapping the parent class 72 to the class 82. Furthermore, mapping module 37 may use any other appropriate technique for mapping one or more classes 72 to one or more classes 82. Translation tool 36 and mapping module 37 may be implemented as any appropriate combination of software

020431.0843

and/or hardware associated with GCD server 40 or with any other appropriate component of system 10.

After the leaf classes 72 of schema 70 have been mapped to classes 82 of schema 80, an ontology may be generated for classes 82 based on the ontology of the leaf classes 72 mapped to classes 82. This ontology creation process may be performed automatically by ontology generation module 38 of translation tool 36. As with mapping module 37, ontology generation module 38 may be implemented as any appropriate combination of software and/or hardware associated with GCD server 40 or with any other appropriate component of system 10. Furthermore, mapping module 37 and ontology generation module 38 may be associated with and executed by the same or by different computers. Ontology generation module 38 creates an ontology for a class 82 by determining the ontology of each leaf class 72 that was mapped to the class 82. The ontology for class 82 is then defined as the intersection of the ontologies of the classes 72 that were mapped to class 82. If a single class 72 was mapped to class 82, the ontology of class 82 may be the ontology of the single class 72. As an example, referring to FIGURE 4, assume that the "Open Sea" and "Sealed" leaf classes 72 (which are included in a "Marine" parent class 72 which is included in a "Batteries" parent class 72) are mapped to a "Batteries" class 82 (which is included in a "Electrical Parts" parent class 82 which is included in a "Marine" parent class 82). Since "Batteries" class 82 does not include specific classes 82 for "open sea" and "sealed" marine batteries, both the "Open Sea" and "Sealed" leaf classes 72 may be mapped to "Batteries" class 82. Therefore, "Batteries" class 82 may include the common attributes from the ontologies of these leaf classes 72.

As described above, the product ontology of a particular class 72 includes the product attributes associated with the class 72 plus the product attributes associated with each of the parent classes 72 of the class 72 (the product attributes associated with each class are indicated in brackets next to the class name in FIGURE 4). Therefore, the ontology associated with "Open Sea" class 72 is as follows: <*voltage, application, type, size, temp*> (assuming that "Batteries" class 72 has no parent class 72 having associated product attributes). Similarly, the ontology associated with "Sealed" class 72 is as follows: <*voltage, application, type, size, gas*>. The new

100-083625-062857-000000000000

ontology of “Batteries” class 82 may then be the intersection of these ontologies, which is as follows: <*voltage, application, type, size*>.

The product attributes that are not included in the intersection of the ontologies of the classes 72 mapped to a particular class 82 (“*temp*” and “*gas*” in the above example) may be used to create subclasses 82 of the particular class (and the pointers associated with the corresponding class 72 may be associated with each subclass 82) or the product attributes may not be included in the ontology of any class 82. Alternatively, the ontology of a particular class 82 may be created from the union of the ontologies of the classes 72 mapped to the class 82. However, in such a case, not all of the products associated with the class 82 (which were associated with the corresponding classes 72) will have associated values for each of the product attributes. Furthermore, any other appropriate technique may be used to create an ontology for a class 82 from the ontologies of classes 72.

After ontologies have been generated for the classes 82 to which classes 72 were associated, there may be classes 82 having the same parent class 82 that have common product attributes in their ontologies. For example, “Batteries” class 82 may have a generated ontology of <*voltage, application, type, size*> and the other classes 82 included in “Electrical Parts” class 82 may also have generated ontologies. The ontology for “Electrical Parts” class 82 may be formed from the intersection of these ontologies. For example, if all the ontologies of the classes 82 included in “Electrical Parts” class 82 include “*voltage*” and “*application*” as attributes, then these two attributes may form the ontology for “Electrical Parts” class 82. These two attributes may then be removed from the attributes associated with the classes 82 under the “Electrical Parts” class 82 since these classes 82 by definition include the attributes of “Electrical Parts” class 82 in their ontologies.

This process may be repeated for each class 82 of schema 80. For example, an ontology may be created for “Marine” class 82 from the intersection (if any) of the ontologies associated with the classes 82 included in “Marine” class 82 (such as “Electrical Parts” class 82). Therefore, in summary, each leaf class 72 of schema 70 is mapped to the most appropriate class or classes 82 of schema 80 and an ontology is created for these classes 82 from the associated mapped classes 72. Then based on the relationship between classes 82 for which an ontology has been generated and the

DRAFTING DATE: 06/23/2023  
PAGES: 10

other classes 82 in schema 80, ontologies may also be generated for these other classes 82, as described above, until all appropriate classes 82 have an associated ontology mapped from the ontology of classes 72 of schema 70.

FIGURE 5 illustrates an example method for translating between different schemas. The method begins at step 160 when mapping module 37 of translation tool 36 (or any other appropriate component used for schema translation) receives information regarding a source schema (such as schema 70) that is to be translated to a target schema (such as schema 80). This information may include, but is not limited to, the taxonomy and ontology of the source schema, the pointers to seller databases 32 and/or repository 34 associated with the classes of the source schema, and the taxonomy of the target schema. Mapping module 37 may be associated with GCD server 40 so that the information regarding a source or target schema associated with GCD 42 may be easily shared with mapping module 37. At step 162, mapping module 37 may generate a graphical representation of the taxonomy of the source and target schemas for presentation to a user. For example, mapping module 37 may generate a tree structure (similar to directory structure 44) to identify the hierarchy of classes that form the taxonomies. Mapping module 37 may communicate the graphical representation of the taxonomies to a user as a web page or other graphical representation using network 12. Mapping module 37 may also present information regarding the taxonomy of the source and target schemas in any other suitable form and using any other suitable communication technique.

Mapping module 37 receives instructions at step 164 from the user regarding the mapping of classes from the source schema to the target schema. For example, mapping module 37 may receive a series of communications from a user in response to the user “dragging and dropping” one or more classes from the source schema (“source classes”) to one or more classes of the target schema (“target classes”). Any other appropriate instructions from the user regarding the mapping of classes may also be used. At step 166, mapping module 37 (or ontology generation module 38) associates the ontology of each source class with its associated target class or classes. Mapping module 37 also associates the pointers associated with each source class to the associated target class at step 168. Therefore, if a buyer 20 selects a particular target class and performs a search for products categorized in that class, the seller

TOP SECRET//  
REF ID: A65280

databases 32 and/or repository 34 including product data for these products will be searched.

At step 170, ontology generation module 38 generates an ontology for the target classes from the intersection of the ontologies of the source classes associated with each target class, as described above. Ontology generation module 38 may receive any required information regarding the mappings and the ontologies from mapping module 37 or data storage associated with translation tool 36. Ontology generation module 38 also generates, at step 172, an ontology for the parent classes of the target classes from the intersection of the ontologies of the child classes of each parent class, as described above. At step 174, ontology generation module 38 generates ontologies for the parent classes of the classes for which ontologies were created at step 172 (from the intersection of the child class ontologies) and also for all appropriate classes above these classes in the hierarchy of the taxonomy until an ontology has been so generated for all appropriate classes in the target schema, at which point the method ends.

As described above, one issue associated with the use of GCD 42 is that GCD 42 may use a schema that is not desired by a particular buyer 20 (for the example, the buyer 20 may desire the use of a schema that is tailored to the buyer's industry). However, as described above, this issue may be addressed by translating a schema provided by GCD 42 into the desired schema. Another issue associated with the use of GCD 42 is that since various types of seller databases 32 are associated with GCD 42, even though these databases 32 may include product data for the same type of product (for example, felt-tip pens), the databases 32 may identify the products using different attribute values, use different names for the same product attribute value, and/or quantify or distinguish product attribute values differently (using different units of measurement, for example). The same may be true for seller data that may be contained in databases 32.

For one or more of these reasons, the seller's product data may not be properly associated with GCD 42 and seller 30 may be disadvantaged during the matching phase of a transaction. For example, if the product ontology associated with pens class 58 in directory structure 44 includes ink color as a product attribute and seller 30 does not have this information in its product data or does not refer to this information

as “ink color” in its database 32, then a search conducted using GCD 42 for pens having a particular ink color may not properly identify products in database 32 that meet the search criteria. Alternatively, the seller’s products may be identified in the search results, but may be ranked lower in the search results since seller 30 does not provide information about the ink color or does provide the information but does not format the information appropriately for use with GCD 42.

Many of these issues may be solved using techniques that identify product and/or seller data in a seller database 32 and properly associate this data with GCD 42 based on the ontology used in a particular schema of GCD 42. If the ontology of the data that is to be associated with GCD 42 is known and understood, then a mapping may be created (manually or automatically) between the ontologies of the data to be associated and the GCD schema. For example, if the *tip size* attribute in the ontology of “pens” class 58 of directory structure 44 is known to correspond to the values in a *tip width* column of a table of product data to be associated, then this column may be mapped to the *tip size* attribute and/or “pens” class 58. For instance, the *tip width* column may be identified using a pointer or the *tip width* attribute may be associated with the *tip size* attribute in GCD 42 so that searches for particular values of *tip size* will cause searches for particular values of *tip width* in the relevant table. However, if the ontology associated with the data to be associated is not known, a number of techniques may be used to identify data in one or more tables of a seller database 32, or other data source and to associate this data with one or more classes of GCD 42 according to the ontology of a particular schema used by GCD 42. The various techniques may be implemented as software that is included in data association module 39. Data association module 39 may be implemented as any appropriate combination of software and/or hardware operating on one or more computers.

FIGURE 6 illustrates an example method of associating product data with a schema of GCD 42. It should be noted that although the association of product data is described, the following techniques apply equally to seller data and any other appropriate data that may be associated with a schema of GCD 42. The example method includes a series of techniques that may be used to identify and associate data with a schema of GCD 42. Although these techniques are described as being performed in a particular sequence in the example method, these techniques may be

performed in any appropriate sequence and one or more of the techniques may not be used. However, it may be advantageous in certain situations to perform the techniques in order from the simplest technique to the hardest technique so that data that may be identified using simpler techniques, if possible, so that the processing required is minimized.

The example method begins at step 200 where data association module 39 accesses the data (the "target data") to be associated with a schema used by GCD 42 (the "target schema"). Data association module 39 may access the target data by accessing a seller database 32 or other appropriate data source, receiving the target data from an appropriate source (such as a seller 30), or using any other appropriate technique. Data association module 39 may access the target data in response to a request from a seller 30 or other appropriate entity or in response to receiving the target data (for example, from a seller 30). The target data may be stored in a table or any other appropriate format. At step 202, data association module 39 accesses the target schema with which the target data is to be associated. This step may involve determining the taxonomy of classes included in the target schema and the ontology of each class. Alternatively, data association module 39 may only determine the ontology associated with selected classes (for example, the leaf classes) or data association module 39 may determine any other appropriate information to be used in associating the target data with a schema.

As described above, the ontology associated with a class includes the names of attributes associated with the class. Since these attribute names are used to identify attribute values in seller databases 32 and repository 34, these attribute names or similar attribute names may be used to identify the target data. For example, these or similar attribute names may be used as column headings in a table including the target data (for example, like the column headings of table 150). Therefore, data association module 39 attempts at step 204 to identify portions of the target data, such as column headings of a table of target data, that match the names of the attributes included in the ontology of one or more classes of the target schema. As an example, data association module 39 may search the target data for each attribute name associated with the ontologies of the target schema. Data association module 39 identifies the data associated with any matching attribute names (such as the values in a column of

DRAFTING DOCUMENTATION

the target data having a heading matching an attribute name) so that the data may be associated with the appropriate classes of the target schema. Although this association may be performed after step 204 is performed (and after each of the other “techniques” described below are performed), the association of data identified using these techniques is described below as step 218 of the example method.

At step 206, data association module 39 attempts to identify portions of the target data that are similar to the names of the attributes included in the ontology of one or more classes of the target schema. Data association module 39 may use an electronic thesaurus to identify equivalents of the attribute names included in the ontologies of the target schema. For example, data association module 39 may determine that “*point width*” and “*tip thickness*” are equivalents of a “*tip size*” attribute. Data association module 39 may then search the target data for each of the equivalents. If a match with a equivalents is found, data association module 39 identifies the target data associated with the matching equivalent (such as the values in a column identified by the equivalent) so that the data may be associated with classes having an ontology including the attribute name from which the equivalent was derived. If appropriate, the data searched may exclude data that was identified in step 204. Furthermore, data identified using any of the techniques described herein may be excluded from consideration by later executed techniques, if appropriate. Therefore, the amount of data that is analyzed may be reduced as each technique is successively performed.

At step 208, data association module 39 attempts to identify portions of the target data by comparing the target data with the values associated with attributes included in the ontology of one or more classes of the target schema. For example, data association module 39 may determine that the following values are associated with a *tip size* attribute in the ontology of a particular class: “broad”, “medium”, and “fine”. Data association module 39 may then search the target data for this collection of values (for example, a column of data in a table including these attributes). As described above, the attribute values may be stored in seller databases 32 and/or repository 34 and may be identified using pointers associated with the relevant classes. To compare the target data with known attribute values, data association module 39 may access the values for a particular attribute and search for one or more

of these attribute values in the target data. Alternatively, data association module 39 may identify portions of the target data that match known attribute values using any other suitable technique. The portions of the target data (for example, particular columns in a table of target data) that are found to match the values associated with a particular attribute may then be associated with the attribute.

It should be noted that the matching values in the target data may not be unique enough by themselves to associate with an attribute of a particular class. For example, data association module 39 may determine that a column of data is likely values for a *price* attribute, but data association module 39 may not be able to determine from these values alone what the product is that is being priced (and multiple class ontologies may have a *price* attribute). However, if multiple portions of the target data are identified (using one or more of the techniques of the example method) as being potentially associated with attributes in the same class ontology, then data association module 39 may use this combination of information to determine that appropriate classes with which to associate the target data. For example, if data association module 39 determines that one column of the target data includes *price* values and another column in the same table of target data includes values for *tip size*, then data association module 39 may determine that the prices are values for a *price* attribute included in a “pens” class (or any other class including *price* and *tip size* attributes).

Data association module 39 attempts to identify portions of the target data at step 210 by comparing the range of values included in the target data with the ranges of values (for example, a numerical range) associated with attributes included in the ontology of one or more classes of the target schema. For example, if a column in a table of target data includes numerical values in the same range as one or more columns of attribute values in a seller database 32 or repository 34, then data association module 39 may determine that the values in the target data correspond to the particular attribute. As described above, although the range of values may not alone be enough information for data association module 39 to determine the appropriate class with which to associate the data, the range of values may be used in association with other identified portions of the target data to make such a determination. To compare a range of a portion of the target data with the range of

TRANSMISSION RECEIVED

known attribute values, data association module 39 may determine the range of values in a particular portion of the target data (such as the data in a particular column) and search for a similar range in the product data stored in seller databases 32 and repository 34. Alternatively, data association module 39 may identify ranges of portions of the target data that match ranges of known attribute values using any other suitable technique. The portions of the target data (for example, particular columns in a table) that are found to match a range of values associated with a particular attribute may then be associated with the attribute.

At step 212, data association module 39 attempts to identify portions of the target data by comparing symbols included in the target data with symbols associated with attribute values associated with one or more classes of the target schema. As an example only, if a column in a table of target data includes dollar signs or other currency symbols, then data association module 39 may determine that the values in the column correspond to a particular attribute or attributes whose values also include dollar signs or other currency symbols. Alternatively, data association module 39 may be programmed to identify particular symbols as being associated with particular attributes (for example, dollar signs are associated with *price* attribute values). Furthermore, data association module 39 may identify target data at step 212 based on the formatting of the data. As an example only, data may be identified based on the position of a decimal point in values included in a portion of the target data. As described above, although the symbols included in the target data and/or the formatting of the target data may not alone be enough information for data association module 39 to determine an appropriate class with which to associate the data, the symbols or formatting may be used in association with other identified portions of data to make such a determination. The portions of the target data (for example, particular columns in a table) that are found to include symbols and/or formatting associated with a particular attribute or attributes may then be associated with the attribute.

If some of the simpler data identification techniques described above are not effective in identifying all of the target data such that the target data may be associated with one or more classes of the target schema, data association module 39 may use more “advanced” techniques. For example, at step 214 data association

Docket No. 020431.0843

module 39 may attempt to identify portions of the target data using vector space analysis of multiple portions of the target data, such as values in multiple columns of a table including the target data. As an example only, data association module 39 may choose  $n$  columns of the target data and “plot” (not necessarily in a graphical sense, but merely analytically) the values in each column along the axis of one of  $n$  dimensions. For instance, data association module 39 may plot the values in one column along the x-axis of a Cartesian coordinate system, the values in another column along the y-axis, and the values of a third column along the z-axis. A similar plot may be made of attribute values associated with one or more classes. The axes of the target data plot may then be rotated until a point of maximum correlation is reached between the target data and the selected attribute values.

For example, if it has been determined based on previous techniques that the target data in a table is associated with a class categorizing tables, but three columns of target data are still not identified, the above technique may be used to associate the unidentified data with particular columns of attribute values in a seller database 32 or repository 34. For instance, the unidentified target data columns may be a height, width, and length of various dining tables. Using the vector space analysis technique described above, data association module 39 may determine which column is height, width, and length, respectively, by correlating the values in these columns with the attribute values in a table in a seller database 32 or repository 34 including data for dining tables. Furthermore, it will be understood that any other appropriate type and application of vector space analysis may also be used to identify target data.

Another technique that data association module 39 may use at step 216 is a statistical correlation technique. Although such techniques may take many forms, one example of such a technique is determining that one attribute in a particular ontology is mathematically related to another attribute in that ontology. For example, for an ontology associated with a class into which box fans are categorized, the values associated with a *height* attribute and a *width* attribute of the ontology may typically be equal or close to equal and the values associated with a *depth* attribute may be equal to a particular fraction of the *height* and *width* values. Furthermore, the power of a box fan (for example, the value of a *wattage* attribute of the ontology) may be related to the size of the fan (for example, the product of *height* and *width* values may

be related to a *wattage* value using a particular mathematical function). Using these known correlations, data association module 39 may identify similar correlations between corresponding values in columns or other portions of target data and thus determine that these columns of data should be associated with the classes having 5 associated attribute values with similar correlations. Furthermore, it will be understood that any other appropriate statistical correlation techniques may be used. Moreover, although particular techniques have been described above any other suitable techniques for identifying target data so that it may be associated with classes of a schema may also be used and are included within the scope of the present invention.

At step 218, data association module 39 associates the data identified in the previous steps with appropriate classes of the target schema. It should be understood that this association may be performed after each portion of data is identified using a particular technique and/or after all the data identification techniques that are going to be performed have been performed. For example, data association module 39 may identify various portions of the target data as being associated with several possible attributes associated with multiple classes until data association module 39 gathers enough information to determine which class or classes the target data should be associated with. For instance, as described above, data association module 39 may 10 determine that a column of target data is associated some *price* attribute, but may not be able to determine which particular *price* attribute with which to associate the data until more of the target data has been identified (since most class ontologies may 15 typically include a *price* attribute). Furthermore, even though data association module 39 may not be able to associate any individual portion of the data with a particular attribute of a class ontology (for example, all portions of the data could 20 individually be associated with numerous different classes), the combination of the “potential” classes with which each portion of data may be associated may identify the particular class or classes with which the data as a whole is to be associated. For 25 instance, if a first portion of the target data could be associated with either Class A or Class B, a second portion could be associated with either Class B or Class C, and a third portion could either be associated with Class B or Class D, then data association 30 module 39 may determine that the data should be associated with Class B.

Data association module 39 may associate data with one or more classes of the target schema at step 218 using any suitable technique. For example, data association module 39 may generate a pointer that identifies the location of a some or all of the target data and associate this pointer with the appropriate class or classes. Therefore,  
5 when a user performs a search from such a class, the pointers will identify the target data as relevant data to be searched. The pointers may identify all of the target data and be generally associated with a class. Alternatively, the pointers may be specific to certain portions of the target data and be associated with the appropriate attribute of the class ontology, so that the target data will be searched for particular values of that attribute identified by a buyer 20 or other party. Any other appropriate techniques for  
10 associating the target data with appropriate classes may also be used.

At step 220, if data association module 39 has not been able to identify portions of the target data, data association module 39 may communicate or otherwise identify this data to a user, such as a seller 30 with which the data is associated, so that the user may appropriately identify the data as being associated with one or more classes of the target schema. Alternatively, even if data association module 39 has identified which classes portions of the target data should be associated with, data association module 39 may communicate this proposed association to the user to obtain a confirmation from the user. In either case, data association module 39 may receive input from the user regarding the association of the target data (either the class or classes with which particular target data is to be associated or confirmation of an association determined by data association module 39) and data association module 39 may then perform the association described with reference to step 218. Alternatively, any other appropriate component of system 10 may be used to make the  
20 appropriate associations.  
25

FIGURE 7 illustrates an example e-commerce system 10 in further detail. As described above, numerous buyers 20 and sellers 30 may be coupled to GCD server 40 using network 12. Buyers 20 may access server 40 using a web browser or in any other appropriate manner and server 40 may provide buyers 20 with access to GCD 42 using a web server or in any other appropriate manner. Although GCD 42 is shown as being internal to GCD server 40, GCD 42 may be internal or external to GCD server 40, as described above. GCD server 40 may also include hardware  
30

and/or software for implementing one or more GCD interfaces 43. A buyer 20 may access server 40 and use a GCD interface 43 to search or navigate GCD 42 and/or seller databases 32. Information may be communicated between buyers 20, sellers 30, and GCD 42 using hypertext transport protocol (HTTP), extensible markup language (XML), simple object access protocol (SOAP), or any other suitable communication technique. Each buyer 20 and seller 30 may be issued a unique identifier so that the participants in a transaction facilitated by GCD 42 may be identified. Each buyer 20 and seller 30 may also be assigned a role with respect to a transaction. As described above, a buyer 20 in one transaction may be a seller 30 in another transaction, and vice versa.

In an example transaction, a buyer 20 may access a GCD interface 43 and perform a search of GCD 42. GCD interface 43 may allow buyer 20 to both navigate or "browse" the classes of GCD 42 and to search for a particular class or classes. For example, buyer 20 may either navigate GCD 42 to find a class into which pens are categorized or buyer 20 may search GCD 42 for class names including the word "pen." Any other suitable methods for identifying a particular class may also be used. When buyer 20 has located the appropriate class for the product buyer 20 desires, buyer 20 may then request a listing of products in that class matching certain product attribute values. For example, if buyer 20 is browsing felt-tip pens class 60b, buyer 20 may request all products in class 60b (felt-tip pens) that have red ink and a fine tip and that are sold by a seller 30 located in the United States.

A search interface 45, or any other appropriate component of GCD server 40, may facilitate such a request by searching or requesting searches of repository 34 and/or seller databases 32 identified by one or more pointers associated with felt-tip pens class 60b. As described above, some of these pointers may have been generated using data association module 39, which may be integral with or separate from GCD server 40. Search interface 45 may provide buyer 20 a search form in which to enter one or more search criteria. The types of search criteria that may be used may be identified in the search form or buyer may be allowed to perform a general search of databases 32 and/or repository 34 for certain terms. For example, search interface 45 may provide buyer 20 with a search form tailored for class 60b that includes fields where buyer 20 can specify a desired ink color, tip thickness, or any other appropriate

product-related or seller-related criteria. In one embodiment, the fields of the search form correspond to some or all of the product attributes within the product ontology and/or seller attributes within the seller ontology corresponding to the product class that has been selected, and buyer 20 may enter values for the product attributes and seller attributes in the corresponding search form fields. In lieu of a search form, search interface 45 may instead provide a single field where buyer can enter in desired search terms, such as “red” and “fine” (multiple search terms may be entered using Boolean operators or any other appropriate technique).

Search interface 45, or any other appropriate component of GCD server 40, may also facilitate search requests by accessing a buyer profile for buyer 20 containing information compiled from previous search requests made by buyer 20, previous e-commerce transactions involving buyer 20, or other events or actions on the part of buyer 20. For example, a buyer profile may contain a list of sellers 30 matching seller attribute values that buyer 20 may want. Such a list may be compiled from the results of previous searches by buyer 20. Search interface 45 may access the profile for buyer 20 for any suitable purpose. In one embodiment, search interface 45 may access the profile for buyer 20 to automatically generate search criteria, such as product attribute values and/or seller attribute values, for a search. Search interface 45 may also access the profile for buyer 20 to limit its search for products matching product attribute values provided by buyer 20 (or generated automatically) to databases 32 associated with sellers 30 known to match seller attribute values that buyer 20 may want (and/or data in repository 34 associated with such sellers 30).

Based on search criteria provided by buyer 20 or automatically generated, search interface 45 may communicate a query to the appropriate seller database(s) 32 and/or repository 34 requesting that databases 32 and/or repository 34 each return a listing of all products (including associated product data and/or seller data) that meet the search criteria. Databases 32 and/or repository 34 may also return data relating to attribute values that were not included in the search criteria. For example, databases 32 may return a price and availability of products that meet the search criteria even if price and availability were not search criteria. The responses to the queries of databases 32 and/or repository 34 may be displayed to buyer 20 in any appropriate manner. For example, the products may be listed in order of relevance to the search

criteria according to any suitable matching criteria. Furthermore, GCD 42 may reorder the product listing based on a request from buyer 20. For example, buyer 20 may request that the matching products be listed in order from least expensive to most expensive. Alternatively, the search results may be communicated directly to buyer 5 20 from databases 32 and/or repository 34.

Buyer 20 may select a product from the product listing to indicate a desire to initiate a transaction regarding the product, such as a purchase of the product. On such a selection, GCD 42 may communicate a repository identifier (RID) identifying the selected seller 30 and a globally unique identifier (GUID) for the product to buyer 10 20. For example, an RID may be the network address (such as an IP address) of a seller network node 30 or may be associated with the network address in a table (in which case GCD 42 may use the RID to look up the associated network address and then communicate the network address to buyer 20). Buyer may access the seller 30 using the RID (or network address) and request a transaction regarding the product using the GUID. GCD 42 may even provide a link including a URL of a web site associated with the seller 30 or may provide another appropriate method for buyer 20 to be connected to seller 20. Although only a single example arrow (between buyer 20n and seller 30n) is shown to illustrate communication between buyers 20 and sellers 30, it should be understood that any buyer 20 may communicate with any seller 30 to conduct appropriate transactions.

Although the present invention has been described with several embodiments, divers changes, substitutions, variations, alterations, and modifications may be suggested to one skilled in the art, and it is intended that the invention encompass all such changes, substitutions, variations, alterations, and modifications falling within the spirit and scope of the appended claims.

DO NOT FILE THIS PAGE